Real-Time 3D reconstruction from single-photon lidar data using plug-and-play point cloud denoisers Supplementary Material

Julián Tachella[1], Yoann Altmann[1], Nicolas Mellado[2], Aongus McCarthy[1], Rachael Tobin[1], Gerald S. Buller[1], Jean-Yves Tourneret[3] and Stephen McLaughlin[1]

[1]School of Engineering and Physical Sciences, Heriot-Watt University, Edinburgh, UK.

[2]IRIT, CNRS, University of Toulouse, Toulouse, France

[3]ENSEEIHT-IRIT-TeSA, University of Toulouse, Toulouse, France.

November 19, 2019

Supplementary Note 1: Details of the 3D reconstruction algorithm

The 3D reconstruction task consists of recovering depth $\boldsymbol{T}$ and intensity $\boldsymbol{R}$ information from the lidar data $\boldsymbol{Z} \in \mathbb{Z}_+^{N_r \times N_c \times T}$, where the photons recorded in pixel (i, j) and histogram bin t are denoted by $z_{i,j,t}$. However, the background image $\boldsymbol{B} \in \mathbb{R}_+^{N_r \times N_c}$ should also be estimated from the data, as it is generally *a priori* unknown and it has a strong impact on the estimation of $\boldsymbol{T}$ and $\boldsymbol{R}$. Consequently, we also estimate $\boldsymbol{B}$ in addition to the depth and intensity profiles. In the single-surface per pixel setting, $\boldsymbol{T} \in \mathbb{R}^{N_r N_c \times 1}$ and $\boldsymbol{R} \in \mathbb{R}_+^{N_r N_c \times 1}$ are vectorized images of fixed size, whereas in the multiple-surface per pixel setting $\boldsymbol{T}$ and $\boldsymbol{R}$ are sets of $N_{\boldsymbol{\Phi}}$ points, which is *a priori* unknown.

Previous 3D reconstruction algorithms

In a fixed dimensional setting (fixed number of points) and assuming one surface (point) per pixel and negligible background levels, the maximum likelihood (ML) solution is

$$(\hat{\boldsymbol{T}}, \hat{\boldsymbol{R}}) = \arg\min_{\boldsymbol{T},\boldsymbol{R}} g\left(\boldsymbol{T}, \boldsymbol{R}, \boldsymbol{B} = \boldsymbol{0}\right) \tag{1}$$

which corresponds to cross-correlating the data with $\log h_{i,j}(t)$ and finding the delay leading to the maximum correlation for each pixel. When the number of photons per pixel is low, or when the single object per pixel assumption does not hold or when the background levels are not negligible, the ML solution does

not provide reliable estimates. These estimates can be improved by considering *a priori* information on the structure of $\boldsymbol{\Phi}$ and $\boldsymbol{B}$. One approach, referred to as penalized maximum-likelihood (PML), introduces additive regularization terms $\rho_{\boldsymbol{T}}(\boldsymbol{T})$, $\rho_{\boldsymbol{R}}(\boldsymbol{R})$ and $\rho_{\boldsymbol{B}}(\boldsymbol{B})$ to enforce more structured solutions, that is

$$(\hat{\boldsymbol{T}}, \hat{\boldsymbol{R}}, \hat{\boldsymbol{B}}) = \arg\min_{\boldsymbol{T},\boldsymbol{R},\boldsymbol{B}} g\left(\boldsymbol{T}, \boldsymbol{R}, \boldsymbol{B}\right) + \lambda_{\boldsymbol{T}}\rho_{\boldsymbol{T}}(\boldsymbol{T}) + \lambda_{\boldsymbol{R}}\rho_{\boldsymbol{R}}(\boldsymbol{R}) + \lambda_{\boldsymbol{B}}\rho_{\boldsymbol{B}}(\boldsymbol{B}) \tag{2}$$

where $\lambda_{\boldsymbol{T}}$, $\lambda_{\boldsymbol{M}}$ and $\lambda_{\boldsymbol{B}}$ are hyperparameters controlling the amount of regularization of the point cloud and background respectively. Following a Bayesian viewpoint, the regularization terms $\rho_{\boldsymbol{T}}(\boldsymbol{T})$, $\rho_{\boldsymbol{R}}(\boldsymbol{R})$ and $\rho_{\boldsymbol{B}}(\boldsymbol{B})$ can be seen as the negative log-prior distributions of the point cloud and background levels, respectively. Under the following assumptions:

1. only one surface per pixel, which reduces to fixing the total number of points to $N_{\boldsymbol{\Phi}} = N_r N_c$,
2. negligible background levels (or removed by a preprocessing step) as in [1, 2] (also equating to $\rho_{\boldsymbol{B}}(\boldsymbol{B}) = 0$)
3. and convex regularization terms $\rho_{\boldsymbol{T}}(\boldsymbol{T})$ and $\rho_{\boldsymbol{R}}(\boldsymbol{R})$,

Problem (2) is convex and has a unique minimizer and it is usually solved by SPIRAL [3] or ADMM [4], which take into account the non-Lipschitz globality of $\nabla_{\boldsymbol{R}} g\left(\boldsymbol{T}, \boldsymbol{R}, \boldsymbol{B}\right)$. However, these assumptions can be too restrictive for practical implementation, as they do not allow for a variable number of surfaces per pixel. Moreover, the depth and intensity regularizations are decoupled, which hinders any improvement of the intensity estimates by using depth information.

In the multiple-surface per pixel scenario, the algorithms investigated in [5] and [6] by-pass the problem related to the unknown number of points by estimating a vectorized data cube of intensities $\boldsymbol{R} \in \mathbb{R}^{N_r \times N_c \times T}$, where the active depths $\boldsymbol{T}$ are implicitly given by the non-zero entries of $\boldsymbol{R}$. Convex priors are then assigned to $\boldsymbol{R}$, such that (2) is convex and has a unique minimiser. However, this formulation presents disadvantages:

1. The estimated values $\boldsymbol{R}$ are generally not sparse enough (over-estimation of the number of points) and the gradient involves a dense computation over the complete cube, implying that these algorithms can come with a high computational complexity.
2. The TV-based regularization term $\rho_{\boldsymbol{R}}(\boldsymbol{R})$ promotes volumetric smoothness, which generally results in poor reconstruction quality and the need of empirical post-processing steps, as the reconstructed surfaces should be manifolds.

The algorithm presented in [7] proposes a model based on spatial point processes to promote manifolds, which improves the results of [5] and [6]. However, since the reconstruction is performed via a reversible jump MCMC algorithm, the method has a intrinsically sequential structure, which is difficult to be parallelized, making the algorithm not well adapted for real-time processing.

Novel reconstruction algorithm

In this work, we avoid the issues induced by the high dimensionality of the intensity parameters involved in the problem formulation adopted in [5] and [6], while allowing for a variable number of surfaces per pixel. More precisely, here $\boldsymbol{T}$ and $\boldsymbol{R}$ are sets of variable size $N_{\boldsymbol{\Phi}}$.

Reparametrization

In a similar fashion to other optimization algorithms assuming Poisson observation noise [8, 9], we introduce the transformation

$$m_n = \log r_n \quad \forall n = 1, \ldots, N_{\boldsymbol{\Phi}} \tag{3}$$

and fix a maximum intensity $m_n \in (-\infty, \log r_{\max}]$. This change of variables and additional constraint ensure that the likelihood remains globally Lipshitz differentiable with respect to $\boldsymbol{R}$. The vectorized set of log-intensity values is denoted by $\boldsymbol{M} = [m_1, \ldots, m_{N_\Phi}]^T$. Analogously, we estimate the log-background levels, i.e., $l_{i,j} = \log b_{i,j}$, denoting the vectorized log-background image as $\boldsymbol{L} = [l_1, \ldots, l_{N_r N_c}]^T$. The resulting negative log-likelihood function under this parametrization is

$$g\left(\boldsymbol{T}, \boldsymbol{M}, \boldsymbol{L}\right) = \sum_{i=1}^{N_c} \sum_{j=1}^{N_r} \sum_{t=1}^{T} z_{i,j,t} \log \left(\sum_{\mathcal{N}_{i,j}} g_{i,j} e^{m_n} h_{i,j}(t - t_n) + g_{i,j} e^{l_{i,j}} \right) - g_{i,j} e^{l_{i,j}} T - \sum_{\mathcal{N}_{i,j}} g_{i,j} e^{m_n} \tag{4}$$

under the assumption of a normalized impulse response, i.e., $\sum_{t=1}^{T} h_{i,j}(t) = 1$ for all the pixels (i, j).

Proximal gradient steps

To solve the general problem in (2), we follow the structure of PALM [10]: the proposed algorithm alternates between the optimization of three blocks of variables ($\boldsymbol{T}$, $\boldsymbol{M}$ and $\boldsymbol{L}$), applying a proximal gradient update on each step, i.e.,

$$\begin{cases} \tilde{\boldsymbol{T}} & \leftarrow \boldsymbol{T}^s - \mu_t^s \nabla_{\boldsymbol{T}} g\left(\boldsymbol{T}^s, \boldsymbol{M}^s, \boldsymbol{L}^s\right) \\ \boldsymbol{T}^{s+1} & \leftarrow \arg\min_{\boldsymbol{T}} \lambda_{\boldsymbol{T}} \rho_{\boldsymbol{T}}(\boldsymbol{T}) + \frac{1}{2\mu_t^s} ||\boldsymbol{T} - \tilde{\boldsymbol{T}}||_2^2 \end{cases} \tag{5}$$

$$\begin{cases} \tilde{\boldsymbol{M}} & \leftarrow \boldsymbol{M}^s - \mu_m^s \nabla_{\boldsymbol{M}} g\left(\boldsymbol{T}^{s+1}, \boldsymbol{M}^s, \boldsymbol{L}^s\right) \\ \boldsymbol{M}^{s+1} & \leftarrow \arg\min_{\boldsymbol{M}} \lambda_{\boldsymbol{M}} \rho_{\boldsymbol{M}}(\boldsymbol{M}) + \frac{1}{2\mu_m^s} ||\boldsymbol{M} - \tilde{\boldsymbol{M}}||_2^2 \end{cases} \tag{6}$$

and

$$\begin{cases} \tilde{\boldsymbol{L}} & \leftarrow \boldsymbol{L}^s - \mu_b^s \nabla_{\boldsymbol{L}} g\left(\boldsymbol{T}^{s+1}, \boldsymbol{M}^{s+1}, \boldsymbol{L}^s\right) \\ \boldsymbol{L}^{s+1} & \leftarrow \arg\min_{\boldsymbol{L}} \lambda_{\boldsymbol{L}} \rho_{\boldsymbol{B}}(\boldsymbol{B}) + \frac{1}{2\mu_b^s} ||\boldsymbol{L} - \tilde{\boldsymbol{L}}||_2^2 \end{cases} \tag{7}$$

where μ_m, μ_t and μ_b are the step sizes for the log-intensities, depths and background levels respectively. The gradients with respect to the log-intensity, depth and background levels are denoted by $[\nabla_{\boldsymbol{M}} g\left(\boldsymbol{T}, \boldsymbol{M}, \boldsymbol{L}\right)]_n = \partial g\left(\boldsymbol{T}, \boldsymbol{M}, \boldsymbol{L}\right) / \partial m_n$, $[\nabla_{\boldsymbol{T}} g\left(\boldsymbol{T}, \boldsymbol{M}, \boldsymbol{L}\right)]_n = \partial g\left(\boldsymbol{T}, \boldsymbol{M}, \boldsymbol{L}\right) / \partial t_n$ and $[\nabla_{\boldsymbol{L}} g\left(\boldsymbol{T}, \boldsymbol{M}, \boldsymbol{L}\right)]_n = \partial g\left(\boldsymbol{T}, \boldsymbol{M}, \boldsymbol{L}\right) / \partial b_n$.

Depth denoising One of the key contributions of this paper is to extend the ideas introduced for plug-and-play denoising [11] to 3D point clouds, replacing the proximal operator of (5) by the APSS algorithm, i.e.,

$$\boldsymbol{T}^{s+1} \leftarrow \text{APSS}(\tilde{\boldsymbol{T}}). \tag{8}$$

The APSS algorithm fits a continuous surface to the set of points defined by $\boldsymbol{T}$, using spheres as local primitives [12, 13]. The algebraic spheres are parametrized by the vector $\boldsymbol{u} = [u_0, \ldots, u_4]^T$, according to the scalar field $\phi : \mathbb{R}^3 \rightarrow \mathbb{R}$, that is

$$\phi_{\boldsymbol{u}}(\boldsymbol{c}) = [1, \boldsymbol{c}^T, \boldsymbol{c}^T\boldsymbol{c}]\boldsymbol{u}. \tag{9}$$

For each 3D point $\boldsymbol{c}_n = [i, j, t_n]^T$, the local sphere is fitted by minimizing the following problem

$$\arg\min_{\boldsymbol{u}} \sum_{n=1}^{N_{\boldsymbol{\Phi}}} w\left(||\boldsymbol{c}_n - \boldsymbol{c}_{\text{r}}||_{\Sigma}\right) \phi^2_{\boldsymbol{u}}(\boldsymbol{c}_{\text{r}}) \tag{10}$$

where $w(t) = (1 - t^2)^4$ is a smooth compactly supported weight function and $||\boldsymbol{c}||_{\Sigma} = \boldsymbol{c}^T\Sigma\boldsymbol{c}$ is a metric of choice, with Σ a diagonal matrix with positive entries, which controls the degree of low-pass filtering of the surface. In particular, $w(t)$ was chosen with diagonal entries, i.e.,

$$\Sigma = \begin{pmatrix} d_x & 0 & 0 \\ 0 & d_y & 0 \\ 0 & 0 & d_t \end{pmatrix} \tag{11}$$

In all the experiment, we set $d_x = d_y = 1$, such that only the 8 closest neighbouring pixels have strong weights, and d_t to be the minimum distance between two surfaces in the same transverse pixel, which is chosen according to the bin width of the lidar system to have a physical meaning (see also the experimental analysis conducted in 'setting the hyperparameters' below). Interestingly, we chose the same distance as the hard constraint between points in the same pixel in ManiPoP [7]. The fitting is performed in real-world coordinates, using the camera mapping $f(\cdot)$. Fig. 1 illustrates the surface fitting performed by APSS. The implicit definition of the scalar field is evaluated in every pixel with at least 3 neighbours, filling any holes and dilating the existing surfaces. Similarly to the *almost orthogonal* projection described in [14], we repeat the fitting process until there is no significant change in the projected point.

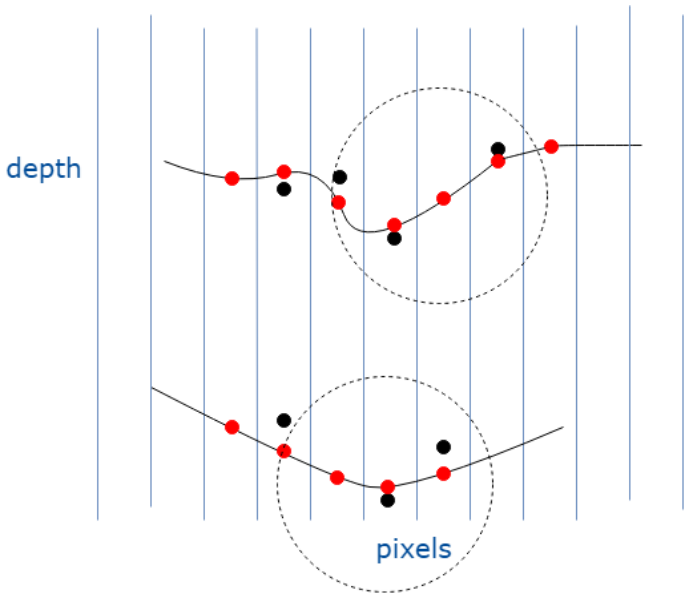


Figure 1: *Example of the APSS denoising step with two surfaces $\mathcal{S}_1$ and $\mathcal{S}_2$ per pixel. The algorithm fits a continuous surface (black line), weighting the 3D points using a kernel defined by an isotropic metric Σ (dashed-line circle). The input points are depicted in black, whereas the output points are red.*

Intensity denoising The proximal operator of the log-intensity update in (6) is replaced by a denoising step using the manifold metrics. In this work, we simply consider a low-pass filter using the nearest neighbours of each point, as in [15]: each log-intensity m_n is updated as

$$m_n^{s+1} = \beta m_n^s + (1-\beta) \sum_{n' \in \mathcal{M}(m_n^s)} \frac{m_{n'}^s}{\# \mathcal{M}(m_n^s)} \tag{12}$$

where β is a coefficient controlling the amount of filtering, $\mathcal{M}(m_n)$ is the set of spatial neighbours m_n and $\# \mathcal{M}(m_n)$ denotes the total number of neighbours. Hence, the proximal step is summarized as

$$\boldsymbol{M}^{s+1} \leftarrow \text{Manifold denoising}(\tilde{\boldsymbol{M}}) \tag{13}$$

More elaborate filters could also be applied, using the manifold metrics defined by the implicit mean least squares surface, as explained in [16]. After the denoising step, we remove the points with intensity r_n lower than a given threshold r_{min}. This step prevents the algorithm from growing surfaces without bounds.

Background denoising The proximal operator used for $\boldsymbol{L}$ depends on the prior assumptions that can be made about the spatial configuration of the spurious detections. Using bistatic raster-scanning systems (e.g., [5]), background counts are not necessarily spatially correlated, thus the proximity operator can be chosen as the identity operator. Using monostatic raster-scanning systems (e.g., [17]) or lidar arrays (e.g., the Princeton Lightwave lidar used in this paper), the background detections appear as a passive image of the

imaged scene. Thus a spatial regularization is useful to improve the background estimates. In this case, we use a Gaussian Markov random field regularization [18], i.e., $\rho_{\boldsymbol{L}}(\boldsymbol{L}) = \boldsymbol{L}^T\boldsymbol{P}\boldsymbol{L}/2$, where $\boldsymbol{P}$ is the Laplacian 2D filter. The proximal operator is thus

$$\boldsymbol{L}^{s+1} \leftarrow (\boldsymbol{I} + \lambda_{\boldsymbol{L}}\mu_b^s\boldsymbol{P})^{-1}\tilde{\boldsymbol{L}} \tag{14}$$

where $\boldsymbol{I}$ is the identity matrix. This denoising step can be quickly computed using the fast Fourier transform (FFT). The proximal operator can also be replaced by an off-the-shelf image denoising algorithm, such as NLM [19] or BM3D [20], at the cost of a higher computational load.

Setting the step sizes

Assuming the number of points is constant, the step sizes at iteration s should verify $\mu_t^s < \frac{1}{L_t^s}$, $\mu_m^s < \frac{1}{L_m^s}$ and $\mu_b^s < \frac{1}{L_b^s}$, where L_t^s, L_m^s and L_b^s are the Lipschitz constants of $\nabla_{\boldsymbol{T}} g\left(\boldsymbol{T}^s, \boldsymbol{M}^s, \boldsymbol{L}^s\right)$, $\nabla_{\boldsymbol{M}} g\left(\boldsymbol{T}^s, \boldsymbol{M}^s, \boldsymbol{L}^s\right)$ and $\nabla_{\boldsymbol{L}} g\left(\boldsymbol{T}^s, \boldsymbol{M}^s, \boldsymbol{L}^s\right)$ respectively [21, 10]. The value of L_t^s can be approximated by assuming that the non-diagonal entries of the Hessian matrix, $\partial g\left(\boldsymbol{T}, \boldsymbol{M}^s, \boldsymbol{L}^s\right)/\partial m_n \partial m_k$ with $k \neq n$, are negligible. Under this approximation, the Lipschitz constant is

$$L_t^s = \max\left\{\frac{\partial^2 g\left(\boldsymbol{T}, \boldsymbol{M}^s, \boldsymbol{L}^s\right)}{\partial t_n^2} \quad n = 1, \ldots, N_{\boldsymbol{\Phi}}\right\}. \tag{15}$$

If the impulse response has a Gaussian shape, i.e., $h_{i,j}(t) \propto \exp(-(t/\sigma)^2/2)$, the partial derivatives can be computed analytically, leading to

$$L_t^s \leq \frac{1}{\sigma^2}\max_{i,j}\sum_{t=1}^{T} z_{i,j,t} \tag{16}$$

which only depends on the width of the impulse response and the maximum number of photons per pixel. Thus, we set a fixed stepsize $\mu_t^s = \mu_t$, dropping the dependence on the iteration s. The values of L_m^s and L_b^s are bounded by the maximum point intensity and background level, that is

$$L_m^s \leq \sum_{t=1}^{T} h_{i,j}(t)\max_n e^{m_n^s} \tag{17}$$

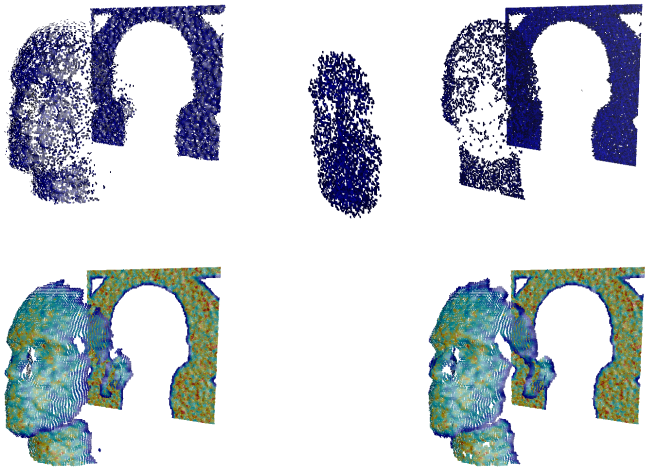
$$L_b^s \leq T\max_{i,j} e^{l_{i,j}}. \tag{18}$$

We bound the maximum intensity, such that L_m^s is upper bounded irrespective of the iteration s. The value of μ_b^s is set to $1/(T\max_{i,j} e^{l_{i,j}^s})$, according to the maximum background level at each iteration.

The aforementioned rules for setting step sizes only guarantee the convergence to a local minimum if the dimension of the problem remains fixed [10]. The overall problem is highly non-convex and changes dimension at each step. Thus, the outcome of the algorithm depends on the initialization. However, as shown in the next section, the algorithm converges to fixed points in practical scenarios and is robust to different initializations.

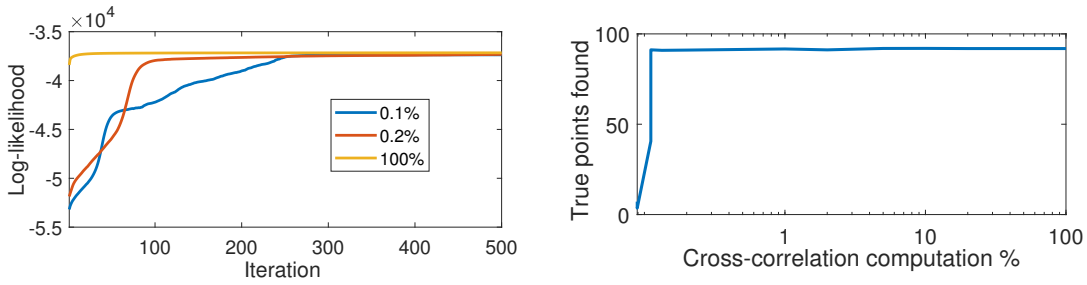
Initialization

The initialization step is designed to provide a coarse estimate, while being fast and easily parallelizable. If at most one surface per pixel is expected, then the classical cross-correlation can be applied. Fig. 2 shows the initialization (top row) and achieved reconstructions (bottom row) for different decimations of the cross-correlation initialization (i.e., the output of the cross-correlation is decimated before finding the bin that realizes the maximum). Decimating the cross-correlation function reduces to considering a reduced number of admissible ranges, which in turn reduces the computational complexity of the initialisation. For instance, Fig. 2 (b) uses only 3 admissible ranges (top subplot). Yet, the algorithm yields the same reconstruction even if 0.11% of the total cross-correlation is computed. As shown in Fig. 3, the algorithm recovers the same amount of true points for a wide range of initialization, converging to the same likelihood value. This approach can be used to further accelerate the algorithm, in the case of one peak per pixel. However, we have proposed a more sophisticated initialisation to handle the more complex scenes.



(a) (b)

Figure 2: *The upper row shows the initialization of the algorithm when computing the* 100% *(a) and* 0.11% *(b) of the total cross-correlation. The bottom row shows the reconstructions obtained after running the proposed reconstruction algorithm.*



(a) (b)

Figure 3: *Left: Value of* $g\left(\boldsymbol{T}, \boldsymbol{M}, \boldsymbol{L}\right)$ *as a function of the iterations of the algorithm for different initializations. Right: Ground-truth points found using different initializations by reducing the computation of the cross-correlation.*

In a general setting where multiple surfaces may be present, we initialize the algorithm with a multi-surface extension of the classic cross-correlation. We propose two different alternatives depending on the sparsity of the recorded histograms:

- Lidar arrays present dense histograms, such that we can use the Anscombe transform [22] to stabilize the variance of the Poisson noise. After the transform, the matching pursuit algorithm [23] is used to find the M most prominent surfaces on each pixel, as summarized in Algorithm 1. The parameter M is user defined and in the experiments presented here, we chose $M = 10$.
- Histograms collected using single-photon lidar systems with high temporal resolution (< 20ps), e.g., raster-scanning systems, generally present a large number of sparsely populated bins, hindering any dense computations using the Anscombe transform. In this case, we find the M most prominent peaks by iteratively using the cross-correlation estimate and removing the photons associated with the peak, as shown in Algorithm 2.

Algorithm 1 Dense VST-MP initialization

Input: lidar waveforms $\boldsymbol{Z}$, maximum number of surfaces per pixel M, $\boldsymbol{z}_{i,j} = [z_{i,j,1}, \ldots, z_{i,j,T}]^T$
Main loop: Process each pixel (i, j) in parallel
$\tilde{\boldsymbol{z}}_{i,j} \leftarrow \text{VST}(\boldsymbol{z}_{i,j})$
$t_1, \ldots, t_M \leftarrow$ Matched Pursuit using $\tilde{\boldsymbol{z}}_{i,j}$ and atoms given by the shifted impulse response $h_{i,j}(t)$
while $s < M$ **do**
    $m_i \leftarrow \log(\sum_{t:h_{i,j}(t-t_i)\neq 0} z_{i,j,t})$
end while
$l_{i,j} = \log(\sum_{\boldsymbol{t}} z_{i,j,t} / \sum_{\boldsymbol{t}} 1)$ where $\boldsymbol{t} = \{t : h_{i,j}(t - t_i) \neq 0 \quad \forall i = 1, \ldots, M\}$
Output: Initial estimates $(\boldsymbol{T}^0, \boldsymbol{M}^0, \boldsymbol{L}^0)$

Algorithm 2 Sparse MP initialization

Input: lidar waveforms $\boldsymbol{Z}$, maximum number of surfaces per pixel M, $\boldsymbol{z}_{i,j} = [z_{i,j,1}, \ldots, z_{i,j,T}]^T$
Main loop: Process each pixel (i, j) in parallel
while $s < M$ **do**
    $t_i \leftarrow$ cross-correlation maximum$(\boldsymbol{z}_{i,j})$
    $m_i \leftarrow \log(\sum_{t:h_{i,j}(t-t_i)\neq 0} z_{i,j,t})$
    $z_{i,j,t} \leftarrow 0 \quad \forall t : h_{i,j}(t - t_i) \neq 0.$
end while
$l_{i,j} = \log(\sum_{\boldsymbol{t}} z_{i,j,t} / \sum_{\boldsymbol{t}} 1)$ where $\boldsymbol{t} = \{t : h_{i,j}(t - t_i) \neq 0 \quad \forall i = 1, \ldots, M\}$
Output: Initial estimates $(\boldsymbol{T}^0, \boldsymbol{M}^0, \boldsymbol{L}^0)$

Setting the hyperparameters

In this section, we study the impact of the hyperparameters on the reconstruction performance, with the aim of providing basic guidelines to select them. We evaluate the performance using the "polystyrene head with backplane" dataset, shown in the main paper. Fig. 4 shows the number of true and false detections as a function of the intensity threshold. As we increase the threshold, the number of true detections decreases monotonically. In contrast, the number of false detections increases exponentially as the threshold tends to zero. The best performing values are between 0.2 and 0.4 photons, coinciding with the reflectivity interval from 5% to 10%. This interval can be used as a guideline for setting r_{min}. The execution time is not affected significantly by the threshold, as the complexity is mostly driven by the (fixed) number of photons.

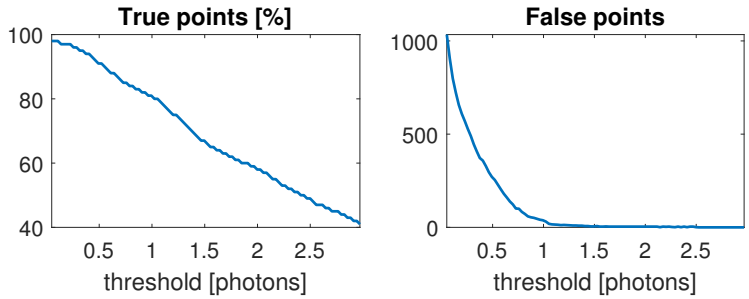


Figure 4: *Reconstruction performance as a function of the intensity threshold for the polystyrene head with backplane. The best performing values have a normalized intensity between* 5% *and* 10%.

The reflectivity update depends on the amount of filtering β in (12), which mostly impacts the intensity estimation. Fig. 5 shows the intensity absolute error (as defined in [24]) as a function of $\beta \in [0, 1]$. Very values of β mean negligible filtering, finding less points and resulting in a larger intensity error. Large values (close to 1) oversmooth the estimates, generating false detections and also resulting in a larger intensity error (this effect is reduced by the very smooth profile of a polystyrene head). Good values for β generally lie in the interval $[0.1, 0.3]$. Note that this interval might vary depending on the number of pixels of the array.

The depth update depends on d_t, the APSS kernel size in the depth direction. Fig. 6 shows the impact

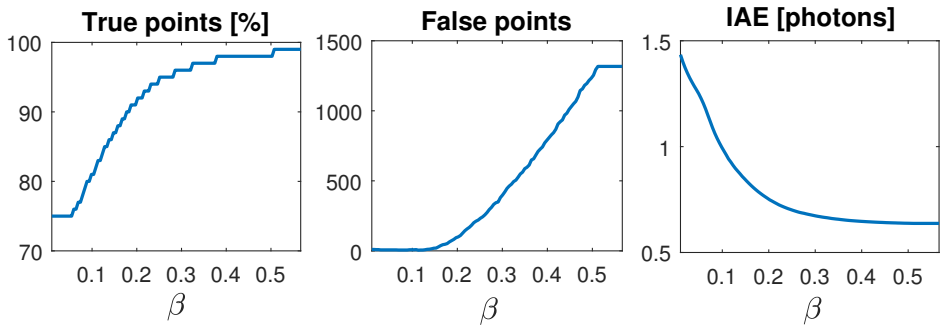


Figure 5: *Effect of the amount of low-pass filtering on the reconstruction quality. Large values oversmooth the estimates, generating false detections and also incurring in a larger intensity error, whereas low values do not impose sufficient spatial correlation, reducing the number of true detections.*

of d_t in terms of true and false detections and mean depth absolute error (DAE). Small values of d_t result in poor reconstructions, as the kernel is too small to correlate neighbouring points, whereas large values oversmooth the depth estimates and may also mix different surfaces. The best choice lies around 8 and 10, which also has the physical meaning discussed above.

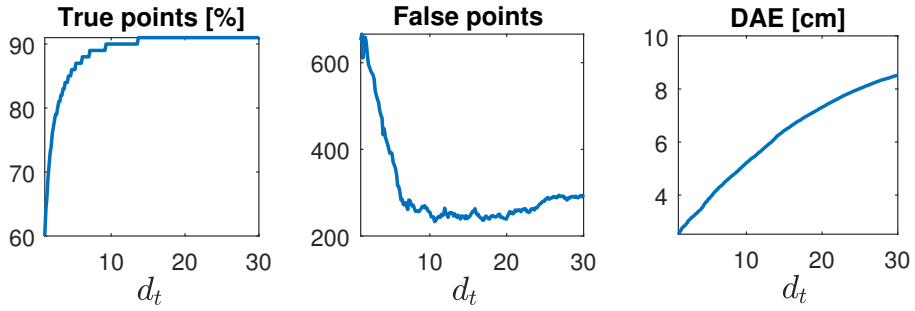


Figure 6: *Effect of the APSS kernel size in the depth direction on the reconstruction quality. Low values fail to correlate neighbouring points, whereas large values oversmooth the depth estimates.*

The background update depends on the hyperparameter $\lambda_{\boldsymbol{L}}$, which controls the degree of correlation between neighbouring background levels. Fig. 7 shows the background estimation performance as a function of $\lambda_{\boldsymbol{L}}$ for the polystyrene head without backplane. While low values of $\lambda_{\boldsymbol{L}}$ do not impose sufficient correlation, large values of $\lambda_{\boldsymbol{L}}$ tend to oversmooth the estimates. While the best choices lie in the interval $[0.5, 2]$, the performance is not very sensitive to bad specifications of $\lambda_{\boldsymbol{L}}$.

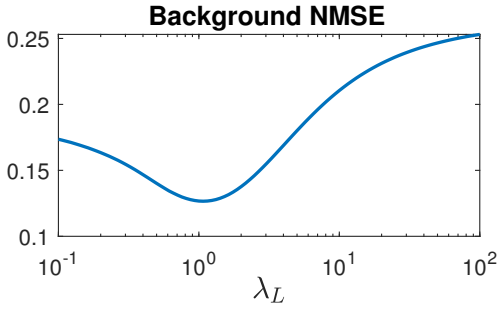


Figure 7: *Effect of the amount of background regularisation λ_L on the estimation of background levels.*

Parallel implementation

Pseudo-code of the full algorithm is presented in Algorithm 3. Our implementation runs completely on a GPU, only exchanging the lidar waveforms and final output with the CPU. The parallel structures of the initialization and main algorithm allow for efficient GPU implementation, as each parallel thread only requires the information of a local subset of photon measurements and 3D points

As the initialization algorithms process every pixel independently, one parallel thread is executed per lidar pixel. The general per-pixel complexity of the dense case is $\mathcal{O}(MT \log T)$, whereas the complexity of the sparse algorithm is $\mathcal{O}(Mk)$, where k is the number of bins with one or more photons.

The gradient and denoising steps of the main algorithm have different parallel implementations. Each of the parallel threads processes one lidar waveform in the gradient steps of (5) and (6), as they can be processed independently of the rest due to the separable structure of the negative log-likelihood. The per-pixel complexity for the depth and log-intensity gradients is $\mathcal{O}(k)$ with k the number of non-zero bins in the compact support of the impulse response centred in the existing points, which is smaller than $\mathcal{O}(T \log T)$ needed for algorithms working on a dense intensity cube such as [5] or [6], especially when the number of histogram bins T is large. The background gradient step in (6) has a complexity of $\mathcal{O}(k)$, where k is the number of active photon pixels in the processed histogram. Both the APSS and intensity denoising steps run one thread per world-coordinates pixels, making use of the shared GPU memory (a gather operation [25]) to efficiently read the information of its neighbours. The main bottleneck of these steps is given by the memory reads during the gather operation, which can be reduced by considering fewer neighbours at the cost of potentially degraded reconstruction. Note that the proposed method has minimal memory requirements. In contrast to alternatives based on convex relaxations [5, 26], which require the storage of a dense 3D cube of intensity estimates of size $\mathcal{O}(N_r N_c T)$, the proposed method only stores the estimated point cloud of size $\mathcal{O}(N_r N_c)$.

The complexity of the algorithm is generally dominated by the gradient steps, which depend on the number of photons (active bins) per pixel. For example, the proposed method might run faster on a large array with few photon detections than a smaller array with densely populated histograms. To illustrate this, consider the execution times of the large raster-scan dataset (13 ms) and the Princeton Lightwave dataset (20 ms). While being significantly smaller, the 32×32 array has dense histograms of 153 bins with non-zero counts. On the other hand, the 141×141 raster scan dataset has a mean photon count of 3 photons per

pixel, hence having approximately 3 active bins per pixel. Hence, the effective data size in the former case is $32 \times 32 \times 153 = 156672$, whereas in the latter is $141 \times 141 \times 3 \times 2 = 119286$ (where the last term in the multiplication is due to the bin number indicator in a sparse representation). The latter data size is smaller than the 32×32 array, hence the faster processing. Moreover, as the algorithm's complexity is driven by the amount of computation within a pixel, it is more intensive to process 153 bins than 4 active bins.

Algorithm 3 Real-time single-photon 3D imaging (RT3D)

Input: lidar waveforms $\boldsymbol{Z}$ and camera parameters $f(\cdot)$
Initialization:
$s \leftarrow 0$
$(\boldsymbol{T}^0, \boldsymbol{M}^0, \boldsymbol{L}^0) \leftarrow$ algorithm 1 (array) or algorithm 2 (raster-scan)
Main loop:
while $s < N_i$ **do**
  $\boldsymbol{T}^{s+1} \leftarrow$ Point cloud denoising $(\boldsymbol{T}^s - \mu_t \nabla_{\boldsymbol{T}} g\left(\boldsymbol{T}^s, \boldsymbol{M}^s, \boldsymbol{L}^s\right))$
  $\boldsymbol{M}^{s+1} \leftarrow$ Manifold denoising $(\boldsymbol{T}^s - \mu_m \nabla_{\boldsymbol{M}} g\left(\boldsymbol{T}^{s+1}, \boldsymbol{M}^s, \boldsymbol{L}^s\right))$
  $\boldsymbol{L}^{s+1} \leftarrow \boldsymbol{L}^s - \mu_b^s \nabla_{\boldsymbol{L}} g\left(\boldsymbol{T}^{s+1}, \boldsymbol{M}^{s+1}, \boldsymbol{L}^s\right)$
  if the lidar system is mono-static **then**
    $\boldsymbol{L}^{s+1} \leftarrow$ Image denoising $(\boldsymbol{L}^{s+1})$
  end if
  $s \leftarrow s + 1$
end while
Output: Final estimates $(\boldsymbol{T}^{N_i}, \boldsymbol{M}^{N_i}, \boldsymbol{L}^{N_i})$

Beyond the APSS denoiser: point cloud denoising alternatives

In this work, we focus on the APSS denoiser to target real-time performance, profiting from the parallel structure and closed-form updates. However, we could imagine other choices with different trade-offs between execution time, memory requirement and reconstruction quality [27]. For example, a straightforward alternative is the simple point set surface (SPSS) denoiser instead of APSS. The proposed method provides a framework to incorporate different types of prior information, avoiding the need to develop specific algorithms for single-photon lidar. As explained in [28], APSS only relies on a local surface smoothness prior, whereas more sophisticated denoisers exploit more complex prior knowledge on the point cloud structure. If we want to capture non-local correlations between point cloud patches, we could use the denoiser in [29], which uses a dictionary learning approach. Higher-level knowledge on the scene, such as the presence of buildings or humans could be also exploited through dedicated denoisers. The algorithm of [30] uses planes to denoise point clouds of building facades, being adapted for remote sensing/outdoor applications. Finally, we could also profit from available 3D data using data-driven denoisers. In this direction, we can use algorithms that fit templates of possible objects [31] or profit from recent advances in graph convolutional neural networks [32], which are specially designed to handle point cloud structures [33, 34].

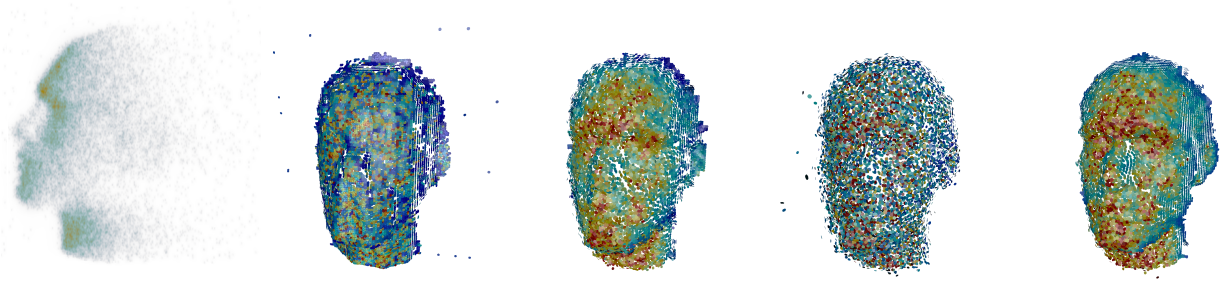
Dataset	Polystyrene head with backplane	Polystyrene head without backplane	Human behind camouflage netting	Mannequin behind scattering object
Source	[37]	[7]	[36]	[5]
N_r	141	141	159	99
N_c	141	141	78	99
T	4613	2500	550	4001
Binning resolution	0.3 mm	0.3 mm	1.2 mm	5.6 mm
Mean phot. per pixel	3.37	1.14	44.6 (long. acq.) 4.5 (short acq.)	45
SBR	13.62	8.14	2.35	8.57
Stand-off distance	40 m	40 m	230 m	4 m
Applicable algorithms	[7, 1] and cross-corr.	[7, 35] and [1] with thres.	[7, 5, 26]	[7, 5, 26]

Table 1: *Summary of the evaluated lidar datasets: N_r and N_c are the number of vertical and horizontal pixels and T is the number of histogram bins. All the datasets were acquired using raster-scan technology.*

Additional results

Comparison with state-of-the-art reconstruction algorithms

We evaluate the proposed method using 4 lidar datasets acquired with different systems, summarised in Table 1. The "polystyrene head with backplane" dataset, shown in the main paper, corresponds to the classical setting with one surface in almost all pixels. The "polystyrene head without backplane" dataset, shown in Fig. 8 and introduced in [35], contains at most one surface per pixel. The "human behind camouflage netting" and "mannequin behind a scattering object", shown in Figs. 9 and 10 and introduced in [36] and [5] respectively, have multiple surfaces per pixel. We compare our results to those obtained with the standard cross-correlation, a state-of-the-art single-surface algorithm [1], 3 multiple-surface reconstruction algorithms SPISTA [5], ℓ_{21}+TV [26] and ManiPoP [7], and a target detection algorithm [35]. Figures 8 to 10 show the 3D reconstructions obtained by the competing algorithms for each dataset, whereas their execution time are presented in Table 2.



Collected photons *Rapp et al. [1]* *ManiPop [7]* *Altmann et al. [35]* *Proposed*

Figure 8: *Comparison of 3D reconstructions achieved by the proposed algorithm and competing methods for the "polystyrene head without backplane" scene. The colour scheme denotes the number of returned photons attributed to each 3D point.*

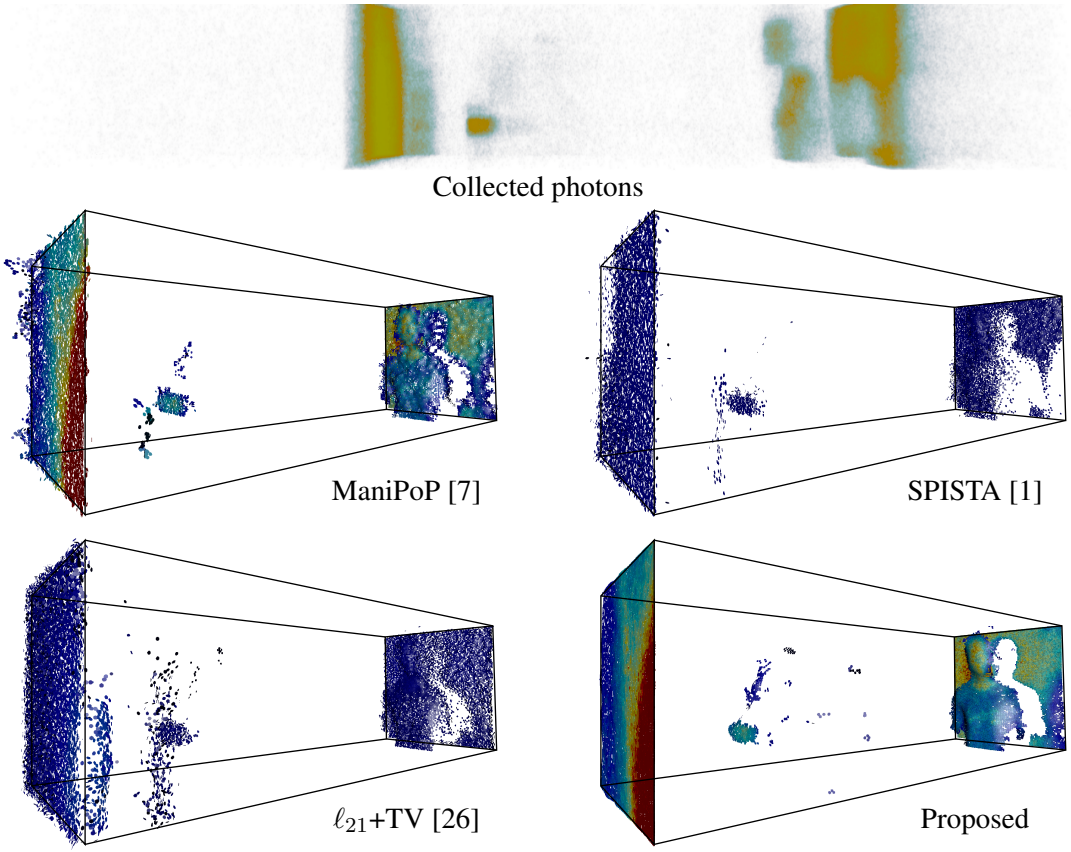
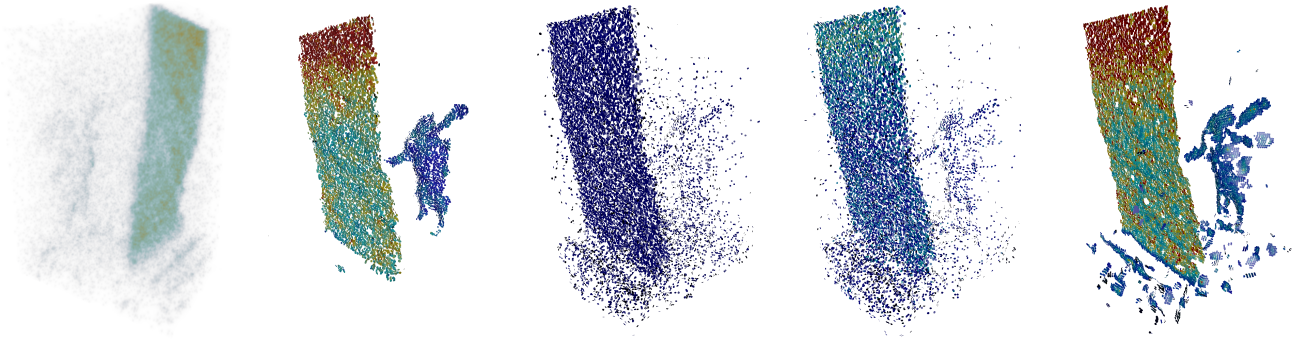
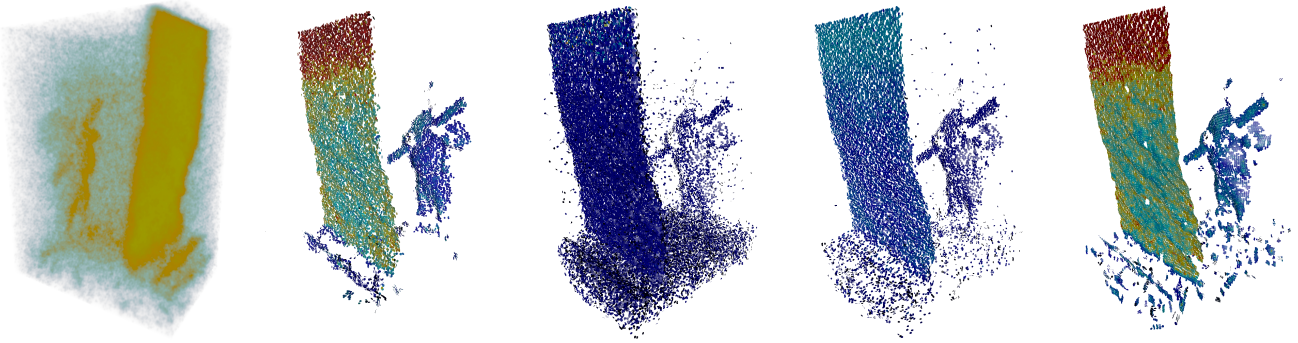


Figure 9: *Comparison of 3D reconstructions achieved by the proposed algorithm and competing methods for the "mannequin behind scattering object" scene.*



(a) *Short acquisition time*



Collected photons *ManiPoP [7]* *SPISTA [5]* *ℓ_{21}+TV [26]* *Proposed*

(b) *Long acquisition time*

Figure 10: *Comparison of 3D reconstructions achieved by the proposed algorithm and competing methods for the "human behind camouflage netting" scene.*

Fig. 11 shows the percentage of true detections and number of false detections as a function of the maximum distance between a ground-truth point and a reconstructed point.

The "polystyrene head without backplane" dataset presents at most one surface per pixel. In this case, if a single-surface per pixel algorithm [1] plus a thresholding step is used, the borders of the target are correlated with spurious detections in pixels without surfaces, yielding relatively poor estimates. The target detection algorithm of [35] takes into account the presence of pixels without any surfaces, but does not promote any correlation between detected points. Both the proposed method and ManiPoP provide good results, correlating only points belonging to the target.

In the "mannequin behind a scattering object" and the "human behind camouflage netting" scenes, [1, 35] or cross-correlation cannot be applied, as they would only reconstruct the first object. In these cases, we evaluate SPISTA, ℓ_{21}+TV, ManiPoP and the proposed method, which can handle multiple surfaces. Again, the best results are obtained by ManiPoP and the proposed algorithm. However, ManiPoP requires an execution time many orders of magnitude higher than the novel method.

	Polystyrene head with backplane	Polystyrene head without backplane	Human behind camouflage netting	Mannequin behind scattering object
Parallel cross-corr	1 ms	1 ms	NA	NA
SPISTA [5]	705 s	3362 s	1279 s (long. acq.) 1212 s (short acq.)	2871 s
ℓ_{21}+TV [26]	201 s	187 s	165 s (long. acq.) 182 s (short acq.)	202 s
Rapp and Goyal [1]	37 s	44 s	NA	NA
Target detection [35]	12 h	12 h	NA	NA
ManiPoP [7]	201 s	181 s	120 s (long. acq.) 102 s (short acq.)	146 s
Proposed method	13 ms	11 ms	27 ms (long. acq.) 15 ms (short acq.)	40 ms

Table 2: *Execution time of the proposed method and other state-of-the-art 3D reconstruction algorithms. Some methods do not provide meaningful results in certain scenes. For such cases, the execution time is not available (NA). The proposed method presents a higher computing time than a parallel implementation of the cross-correlation algorithm (which only applies in the presence of single peaks), but outperforms all the other reconstruction algorithms by a factor of about* $\approx 10^5$.

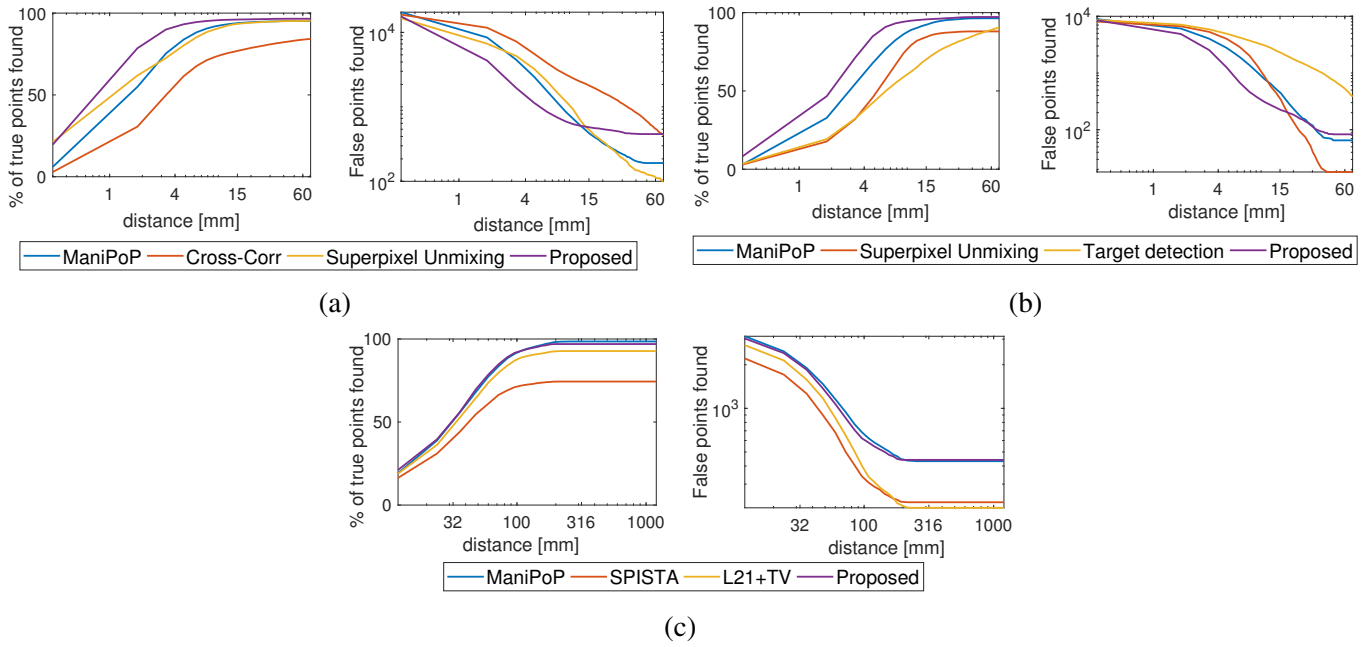


Figure 11: *Number of true and false detections as a function of the maximum admissible distance between a ground truth point and a reconstructed one for (a) head with backplane (b) head without backplane and (c) mannequin behind scattering object.*

Improvements by upsampling in small lidar arrays

The proposed upsampling can bring additional details to the reconstructed objects, improving the estimates of naive upsampling in a post-processing step. Fig. 12 shows the upsampled reconstructions with the proposed method and cross-correlation. The cross-correlation output was upsampled by naively converting each detection into a 3×3 grid of points at the same depth. While the upsampled cross-correlation has a blocky appearance, the proposed method captures additional details in the contours of the 3D target. Note that these contours are not always aligned with the coarse scale.

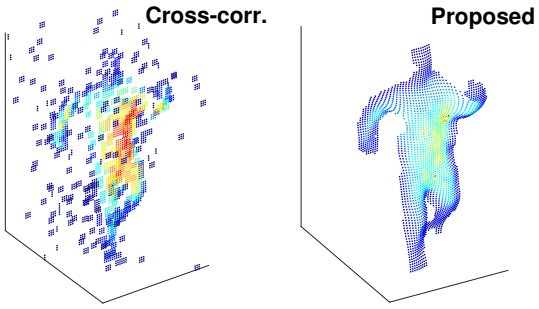


Figure 12: *Comparison of 3D reconstructions of the* 32×32 *lidar array data using cross-correlation and the proposed method. The upsampling strategy of the proposed method brings additional details in the contours of the object, whereas a naive upsampling of the cross-correlation output presents a blocky appearance.*

Operation boundary conditions

Finally, we study the performance of the algorithm as a function of the mean number of photons per pixel and signal-to-background ratio. We generate 100 synthetic lidar cubes for SBR values in $[0.01, 100]$ and mean photons per pixels in $[0.1, 100]$, using the ground truth point cloud, data cube size and impulse response from the "polystyrene head without backplane" dataset. As a baseline, we compare the proposed method with the standard cross-correlation algorithm. To account for the pixels without objects, we post-process the output of cross-correlation by removing points below a normalized intensity of 10%. We consider the number of true and false detections, depth absolute error (only computed for true detections and reconstructions with more than 80% of detected points), intensity absolute error (normalised by the mean signal photon counts to approximately lie between 0 and 1) and background NMSE. The results obtained are gathered in Fig. 13. The proposed method performs well in a wider range of conditions, achieving reconstructions with ≈ 0.1 photons per pixel and up to signal-to-noise background ratio of 0.01 (with 100 photons per pixel or more). Moreover, cross-correlation generates many orders of magnitude more false detections than the new method. Interestingly, the proposed algorithm exhibits a sharper transition in the detection of true points, meaning that, for a given signal-to-background ratio, either none or most of the points will be found depending on the recorded photon count. The new method achieves smaller depth and intensity absolute errors than cross-correlation in all conditions, as it exploits the manifold structure of the scene. The proposed method also achieves a significantly smaller background NMSE, capturing the spatial correlation in the background image.

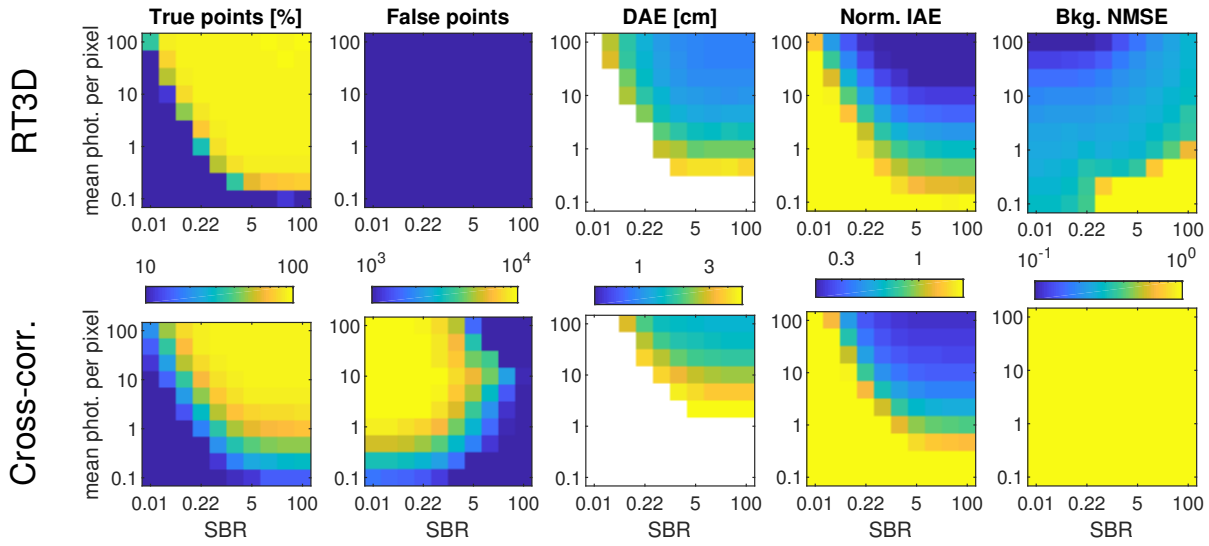


Figure 13: *Comparison of the proposed method and cross-correlation with thresholding in a target detection setting for different SBR and mean photons per pixel values. The depth absolute error is only displayed for reconstructions with more than* 80% *and is left blank otherwise.*

Supplementary methods

3D lidar array

A schematic diagram of the lidar system, which was based on the 32×32 single-photon array Kestrel camera produced by Princeton Lightwave, is shown in Fig. 14. The system was implemented as a bistatic arrangement - the illuminating transmit (laser) channel and the collecting receive (camera) channel were not co-axial - with the centres of the apertures separated by about 125 mm. This configuration was used in order to avoid potential issues that could arise in a co-axial (monostatic) system due to back reflections from the optical components causing damage to the sensitive focal plane array of the Kestrel camera. The bistatic optical configuration meant that a slight re-alignment of the illumination channel, relative to the receive (camera) channel, was required for scenes at different distances from the system. Both the camera and laser were mounted on a single breadboard with the optical setup for the illumination channel mounted on a stage which enabled controlled adjustments to be made to the pitch and yaw of the illuminating beam, so that it could be positioned accurately relative to the field of view of the camera. Another camera (Ninox 640 VIS-SWIR, from Raptor Photonics) was also mounted on the breadboard and used to help align the system to the scene of interest.

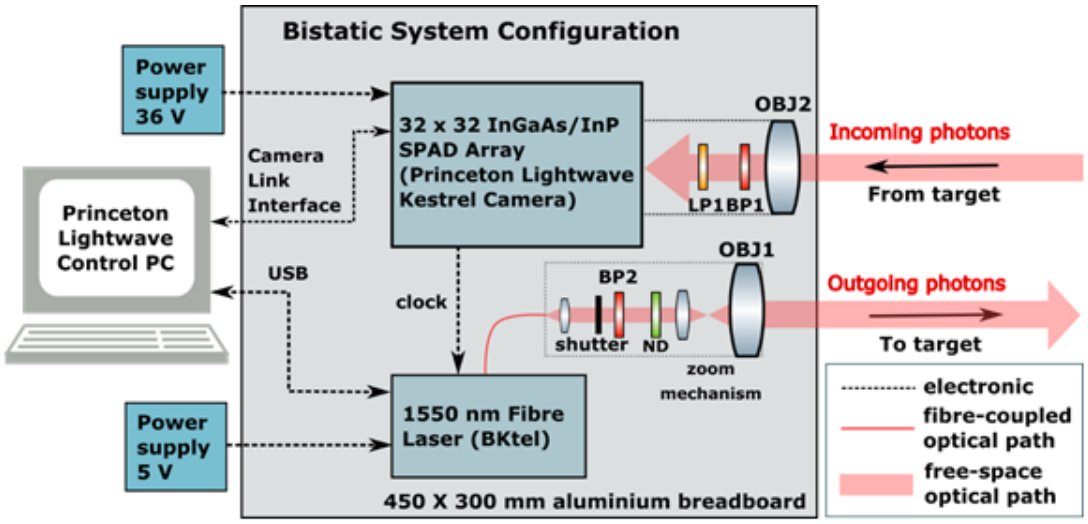


Figure 14: *A schematic diagram of the lidar system showing the key components and configuration of the bistatic imaging system comprising the Princeton Lightwave Kestrel* 32×32 *InGaAs/InP SPAD detector array and the* $\lambda = 1550$ *nm fibre pulsed illumination source. Optical components include: objective lenses (OBJ1, OBJ2)); a neutral density filter (ND); a longpass filter (LP1); and bandpass filters (BP1, BP2). Details of these components are given in the text.*

The operating wavelength for the system was chosen as 1550 nm - this wavelength corresponds to a high transmission window in the atmosphere, with the unwanted contribution from solar background being significantly lower when compared to shorter wavelengths, and it is eye-safe at significantly higher power levels than for wavelengths in the retinal hazard region of the spectrum (which extends from 400 to 1400 nm). The BKtel fibre laser (HFL-240am series) had a central wavelength of 1550 nm and the pulse width was measured to be 413 ps at the operating parameters used in these measurements. It was run at a repetition rate of 150.421 kHz (this clock signal was provided by the Kestrel camera), and the resulting average optical output power was approximately 220 mW (for a laser drive current of 3 A). A neutral density (ND) filter with an optical density of 0.5 and transmission of approximately 32% at $\lambda = 1550$ nm was used to reduce the average optical power level to approximately 70 mW to avoid saturating the sensitive detector. The output fibre from the laser module was connected to a reflective collimation package and the exiting beam was then passed through a 12 nm FWHM bandpass filter with a centre wavelength of 1550 nm in order to remove any amplified spontaneous emission that was present. A beam expander arrangement consisting of a pair of lenses (with effective focal lengths of approximately 10 mm and 75 mm) housed in a zoom mechanism enabled the diameter of the illuminating beam at the scene of interest to be adjusted to match the field of view of the camera.

The Kestrel camera had an InGaAs/InP SPAD detector array and at the operating wavelength of 1550 nm, the elements in the array had a quoted photon detection efficiency of approximately 25% and a measured dark count rate of approximately 320 kcps. The camera was operated in time-of-flight mode and configured to operate with 250 ps timing bins, a gate duration of 40 ns, which corresponds to a total of 160 histogram bins, and was equivalent to a measurement depth range of 6 metres. The camera was operated at a frame rate

of 150.421 kHz (this was close to the expected maximum frame rate of the camera). In order to acquire an accurate instrumental response of the system (i.e., accurate estimations of $h_{i,j}(t)$ and $g_{i,j}$) a long-acquisition measurement of a uniform, cooperative surface (Spectralon, Labsphere, Inc) was made in dark laboratory conditions over a short stand-off distance of 2 metres. Due to a small amount of latency present in the timing electronics of each SPAD detector, the instrumental response of each pixel in the camera array is non-identical. This is taken in to consideration during data reconstruction by using a separate instrumental response for each pixel.

A 500 mm effective focal length lens operating at f/7 (manufactured by Optec, and designed for use in the 900 to 1700 nm wavelength region) was attached to the camera to collect the scattered return photons from the scene. This resulted in a field of view of approximately 2×2 metres at the standoff distance of 320 metres, i.e., each individual pixel covered an area of approximately 65×65 mm. In order to minimise the amount of background light detected, a pair of high performance passive spectral filters was mounted between the rear element of the lens and the sensor of the camera - one was a longpass filter with a cut-on wavelength of 1500 nm, and the other was a 9 nm full width half maximum (FWHM) bandpass filter with a centre wavelength of 1550 nm.

References

[1] J. Rapp and V. K. Goyal, "A few photons among many: Unmixing signal and noise for photon-efficient active imaging," *IEEE Transactions on Computational Imaging*, vol. 3, no. 3, pp. 445–459, 2017.

[2] D. Shin, A. Kirmani, V. K. Goyal, and J. H. Shapiro, "Photon-efficient computational 3-D and reflectivity imaging with single-photon detectors," *IEEE Transactions on Computational Imaging*, vol. 1, no. 2, pp. 112–125, 2015.

[3] Z. T. Harmany, R. F. Marcia, and R. M. Willett, "This is SPIRAL-TAP: Sparse poisson intensity reconstruction algorithms: Theory and practice," *IEEE Transactions on Image Processing*, vol. 21, no. 3, pp. 1084–1096, Mar. 2012.

[4] M. A. T. Figueiredo and J. M. Bioucas-Dias, "Restoration of Poissonian images using alternating direction optimization," *IEEE Transactions on Image Processing*, vol. 19, no. 12, pp. 3133–3145, Dec. 2010.

[5] D. Shin, F. Xu, F. N. Wong, J. H. Shapiro, and V. K. Goyal, "Computational multi-depth single-photon imaging," *Optics express*, vol. 24, no. 3, pp. 1873–1888, 2016.

[6] A. Halimi, Y. Altmann, A. McCarthy, X. Ren, R. Tobin, G. S. Buller, and S. McLaughlin, "Restoration of intensity and depth images constructed using sparse single-photon data," in *Proc. 24th European Signal Processing Conference (EUSIPCO), Budapest, Hungary*, Aug. 2016, pp. 86–90.

[7] J. Tachella, Y. Altmann, X. Ren, A. McCarthy, G. Buller, S. McLaughlin, and J. Tourneret, "Bayesian 3D reconstruction of complex scenes from single-photon lidar data," *SIAM Journal on Imaging Sciences*, vol. 12, no. 1, pp. 521–550, 2019. [Online]. Available: https://doi.org/10.1137/18M1183972

[8] J. Salmon, Z. Harmany, C.-A. Deledalle, and R. Willett, "Poisson noise reduction with non-local PCA," *Journal of Mathematical Imaging and Vision*, vol. 48, no. 2, pp. 279–294, Feb. 2014. [Online]. Available: https://doi.org/10.1007/s10851-013-0435-6

[9] W. Marais and R. Willett, "Proximal-gradient methods for Poisson image reconstruction with BM3D-based regularization," in *Proc. 7th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), Curacao, Dutch Antilles*, Dec. 2017, pp. 1–5.

[10] J. Bolte, S. Sabach, and M. Teboulle, "Proximal alternating linearized minimization for nonconvex and nonsmooth problems," *Mathematical Programming*, vol. 146, no. 1, pp. 459–494, Aug 2014. [Online]. Available: https://doi.org/10.1007/s10107-013-0701-9

[11] S. Sreehari, S. V. Venkatakrishnan, B. Wohlberg, G. T. Buzzard, L. F. Drummy, J. P. Simmons, and C. A. Bouman, "Plug-and-play priors for bright field electron tomography and sparse interpolation," *IEEE Transactions on Computational Imaging*, vol. 2, no. 4, pp. 408–423, Dec. 2016.

[12] G. Guennebaud and M. Gross, "Algebraic point set surfaces," *ACM Trans. Graph.*, vol. 26, no. 3, Jul. 2007. [Online]. Available: http://doi.acm.org/10.1145/1276377.1276406

[13] G. Guennebaud, M. Germann, and M. Gross, "Dynamic sampling and rendering of algebraic point set surfaces," *Computer Graphics Forum*, vol. 27, no. 2, pp. 653–662, 2008. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8659.2008.01163.x

[14] M. Alexa and A. Adamson, "On normals and projection operators for surfaces defined by point sets," in *Proc. Eurographics Conference on Point-Based Graphics*, ser. SPBG'04. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2004, pp. 149–155. [Online]. Available: http://dx.doi.org/10.2312/SPBG/SPBG04/149-155

[15] J. B. Tenenbaum, V. d. Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000. [Online]. Available: http://science.sciencemag.org/content/290/5500/2319

[16] J. Liang and H. Zhao, "Solving partial differential equations on point clouds," *SIAM Journal on Scientific Computing*, vol. 35, no. 3, pp. A1461–A1486, 2013.

[17] A. M. Pawlikowska, A. Halimi, R. A. Lamb, and G. S. Buller, "Single-photon three-dimensional imaging at up to 10 kilometers range," *Opt. Express*, vol. 25, no. 10, pp. 11 919–11 931, May 2017. [Online]. Available: http://www.opticsexpress.org/abstract.cfm?URI=oe-25-10-11919

[18] H. Rue and L. Held, *Gaussian Markov random fields: theory and applications*. CRC press, 2005.

[19] A. Buades, B. Coll, and J. . Morel, "A non-local algorithm for image denoising," in *Proc. Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 2, June 2005, pp. 60–65 vol. 2.

[20] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, Aug. 2007.

[21] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 127–239, 2014. [Online]. Available: http://dx.doi.org/10.1561/2400000003

[22] F. J. Anscombe, "The transformation of Poisson, binomial and negative-binomial data," *Biometrika*, vol. 35, no. 3/4, pp. 246–254, 1948. [Online]. Available: http://www.jstor.org/stable/2332343

[23] S. G. M. and, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, Dec. 1993.

[24] J. Tachella, Y. Altmann, M. Márquez, H. Arguello-Fuentes, J.-Y. Tourneret, and S. McLaughlin, "Bayesian 3D Reconstruction of Subsampled Multispectral Single-photon Lidar Signals," *arXiv e-prints*, p. arXiv:1904.02583, Apr 2019.

[25] J. Sanders and E. Kandrot, *CUDA by example: an introduction to general-purpose GPU programming*. Addison-Wesley Professional, 2010.

[26] A. Halimi, R. Tobin, A. McCarthy, S. McLaughlin, and G. S. Buller, "Restoration of multilayered single-photon 3d lidar images," in *Proc. 25th European Signal Processing Conference (EUSIPCO), Kos Island, Greece*, Aug. 2017, pp. 708–712.

[27] V. Chandrasekaran and M. I. Jordan, "Computational and statistical tradeoffs via convex relaxation," *In Proc. of the National Academy of Sciences*, vol. 110, no. 13, pp. E1181–E1190, 2013. [Online]. Available: https://www.pnas.org/content/110/13/E1181

[28] M. Berger, A. Tagliasacchi, L. M. Seversky, P. Alliez, G. Guennebaud, J. A. Levine, A. Sharf, and C. T. Silva, "A survey of surface reconstruction from point clouds," *Computer Graphics Forum*, vol. 36, no. 1, pp. 301–329, 2017. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.12802

[29] S. Xiong, J. Zhang, J. Zheng, J. Cai, and L. Liu, "Robust surface reconstruction via dictionary learning," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 6, p. 201, 2014.

[30] C.-H. Shen, S.-S. Huang, H. Fu, and S.-M. Hu, "Adaptive partitioning of urban facades," in *ACM Transactions on Graphics (TOG)*, vol. 30, no. 6. ACM, 2011, p. 184.

[31] L. Nan, K. Xie, and A. Sharf, "A search-classify approach for cluttered indoor scene understanding," *ACM Trans. Graph.*, vol. 31, no. 6, pp. 137:1–137:10, Nov. 2012. [Online]. Available: http://doi.acm.org/10.1145/2366145.2366156

[32] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: Going beyond euclidean data," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 18–42, July 2017.

[33] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "PointCNN: Convolution on X-transformed points," in *Advances in Neural Information Processing Systems (NIPS 2018), Montreal, Canada*. Curran Associates, Inc., 2018, pp. 820–830. [Online]. Available: http://papers.nips.cc/paper/7362-pointcnn-convolution-on-x-transformed-points.pdf

[34] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic Graph CNN for Learning on Point Clouds," *arXiv e-prints*, p. arXiv:1801.07829, Jan 2018.

[35] Y. Altmann, X. Ren, A. McCarthy, G. S. Buller, and S. McLaughlin, "Robust Bayesian target detection algorithm for depth imaging from sparse single-photon data," *IEEE Transactions on Computational Imaging*, vol. 2, no. 4, pp. 456–467, Dec. 2016.

[36] R. Tobin, A. Halimi, A. McCarthy, X. Ren, K. J. McEwan, S. McLaughlin, and G. S. Buller, "Long-range depth profiling of camouflaged targets using single-photon detection," *Optical Engineering*, vol. 57, no. 3, pp. 1 – 10 – 10, 2017. [Online]. Available: https://doi.org/10.1117/1.OE.57.3.031303

[37] Y. Altmann, X. Ren, A. McCarthy, G. S. Buller, and S. McLaughlin, "Lidar waveform-based analysis of depth images constructed using sparse single-photon data," *IEEE Transactions on Image Processing*, vol. 25, no. 5, pp. 1935–1946, 2016.